Research Note 83-4

AD A127128

DEVELOPMENT AND DEMONSTRATION OF A LABORATORY TOOL FOR RESEARCH IN THE DESIGN OF GAMES FOR TRAINING OF TROUBLESHOOTING SKILLS

William B. Johnson, William C. Entwistle,
Karen S. Gaddis

University of Illinois at Urbana-Champaign
Institute of Aviation

TRAINING RESEARCH LABORATORY

U. S. Army

Research Institute for the Behavioral and Social Sciences

August 1982

Approved for public release; distribution unlimited.

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER Research Note 83-4	2. GOVT ACCESSION NO. AD-A127128	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Development and Demonstration of a Laboratory Tool for Research in the Design of Games for Training of Troubleshooting Skills		5. TYPE OF REPORT & PERIOD COVERED Final Report May 1982 - August 1982
		6. PERFORMING ORG. REPORT NUMBER ARL-TR82-1 ✓
7. AUTHOR(s) William B. Johnson, William C. Entwistle, and Karen S. Gaddis		8. CONTRACT OR GRANT NUMBER(s) MDA 903-82-M-3924
9. PERFORMING ORGANIZATION NAME AND ADDRESS University of Illinois Institute of Aviation Willard Airport Savoy, IL 61874		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 2Q162722A791
11. CONTROLLING OFFICE NAME AND ADDRESS Army Research Institute for the Behavioral and Social Sciences. 5001 Eisenhower Avenue. Alexandria, Virginia 22333		12. REPORT DATE August 1982
		13. NUMBER OF PAGES
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Troubleshooting
Maintenance Training
PLATO
Games

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

In 1979 the University of Illinois developed a computer-based simulation designed to assist aviation maintenance trainees in acquiring troubleshooting skills. This simulation was named FAULT (Framework for Aiding in the Understanding of Logical Troubleshooting) and was originally designed for the DEC system 10 computer.

The United States Army Research Institute for the Behavioral and Social

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

item #20 - continued

Sciences subsequently contracted with the Aviation Research Laboratory (ARL) of the University of Illinois to develop a FAULT simulation compatible with the PLATO system. The PLATO system is used extensively by the Army for both training and research purposes.

In order to meet the Army's needs, ARL researchers made certain key modifications to the original FAULT simulation which facilitate its use by troubleshooters and Army researchers.

As a training tool, PLATO FAULT simulates troubleshooting decisions in a game format. The object of the game is to find the single failed part as quickly as possible while incurring the minimum expense. The instructions and on-line help provided to the trainee have been redesigned to provide more information in natural language. Throughout the simulation the trainee has available a number of options that allow him to (1) gather information about the problem, (2) act on the information, and (3) receive information about the action. To present a realistic simulation of actual troubleshooting, FAULT assigns dollar charges to certain options and the list of options has been clearly presented to differentiate between options that are free and those that incur costs.

For use in research, the PLATO version of FAULT allows the researcher to create demonstrations and experiments. The version provided to the Army contains one operational technical system (TRUCK ENGINE) for immediate use and the necessary documentation to modify this system and develop additional ones. Furthermore, the programming allows the experimenter to select the problems that will be presented to the troubleshooter. The collection and accessibility of data have also been enhanced: data can be collected and stored simultaneously from numerous PLATO IV terminals and this data can be accessed either on a global level (i.e., average times and costs to problem solution) or on a very specific level (i.e., listing of each action required to solve a problem).

AVIATION RESEARCH LABORATORY

Institute of Aviation

University of Illinois at Urbana-Champaign
Willard Airport
Savoy, Illinois
61874

Technical Report

ARL-TR82-1

August, 1982

DEVELOPMENT AND DEMONSTRATION OF A LABORATORY TOOL FOR RESEARCH IN THE DESIGN OF GAMES FOR TRAINING OF TROUBLESHOOTING SKILLS

William B. Johnson, Ph.D.
William C. Entwistle
Karen S. Gaddis

DESIGN SPECIFICATION AND FINAL REPORT
Contract No. MDA 903-82-M-3924

Prepared for

UNITED STATES ARMY RESEARCH INSTITUTE FOR
THE BEHAVIORAL AND SOCIAL SCIENCES

Executive Summary

This report for the United States Army Research Institute for the Behavioral and Social Sciences, Alexandria, Virginia, fulfills the terms of contract MDA 903-82-M-3924 for the design specification and final report of the computer simulation hereafter referred to as PLATO FAULT.

The report contains a description of PLATO FAULT and includes the appropriate screen copies to demonstrate the simulation to the reader. Also included, predominantly as appendices, is documentation for the researcher.

This report has been reviewed and is approved for publication, August, 1982.

Henry L. Taylor, Acting Head
Aviation Research Laboratory and
Director, Institute of Aviation
University of Illinois at Urbana-Champaign

Contents

I. Introduction

The University of Illinois, primarily at the Coordinated Science Laboratory under Professor William B. Rouse, has been involved in a variety of computer-based, problem-solving research projects. During 1979, under contract MDA 903-79-C-9421, a simulation named FAULT (Framework for Aiding in the Understanding of Logical Troubleshooting) was developed and evaluated. The conceptualization, development, and evaluation of the simulation, which included ten experiments with live subjects, have been reported extensively (Hunt, R.M., 1979; Hunt, R.M. and Rouse, W.B., 1981; Johnson, W.B., 1981; Johnson, W.B. and Rouse, W.B., 1982, b; Johnson, W.B. and Rouse, W.B., 1982, c; Rouse, W.B. and Hunt, R.M., 1982).

The present project had the goal of capitalizing on the previous empirical evaluations to develop a FAULT simulation compatible with the PLATO system (Johnson, 1982). The project had the contractual obligation to develop a simulation with at least one technical system. The simulation had to permit the simultaneous collection and storage of data from numerous PLATO IV terminals. A program had to be written to make the data accessible to the Army Research Institute (ARI) experimenter.

The meeting of May 26, 1982 at ARI with Douglas Bobko, Carolyn Carroll, Michael Drillings, Dick Gebhard, John Hays, Rich Johnson and W.B. Johnson, made it clear that ARI wanted a demonstration tool that could be used conveniently throughout the

extensive Army PLATO system. It was emphasized that the computer code should contain the necessary comments and explanations to facilitate modification by ARI programmers. PLATO FAULT meets or exceeds the contractual specifications.

II. PLATO FAULT

In order to use PLATO FAULT, one must use a hardcopy display of the specific technical system which is being used for the troubleshooting simulation. PLATO FAULT is presently configured with Truck Engine (see Figure 1). We modified the prices and part descriptions of an earlier simulation, Car Engine, in developing Truck Engine.

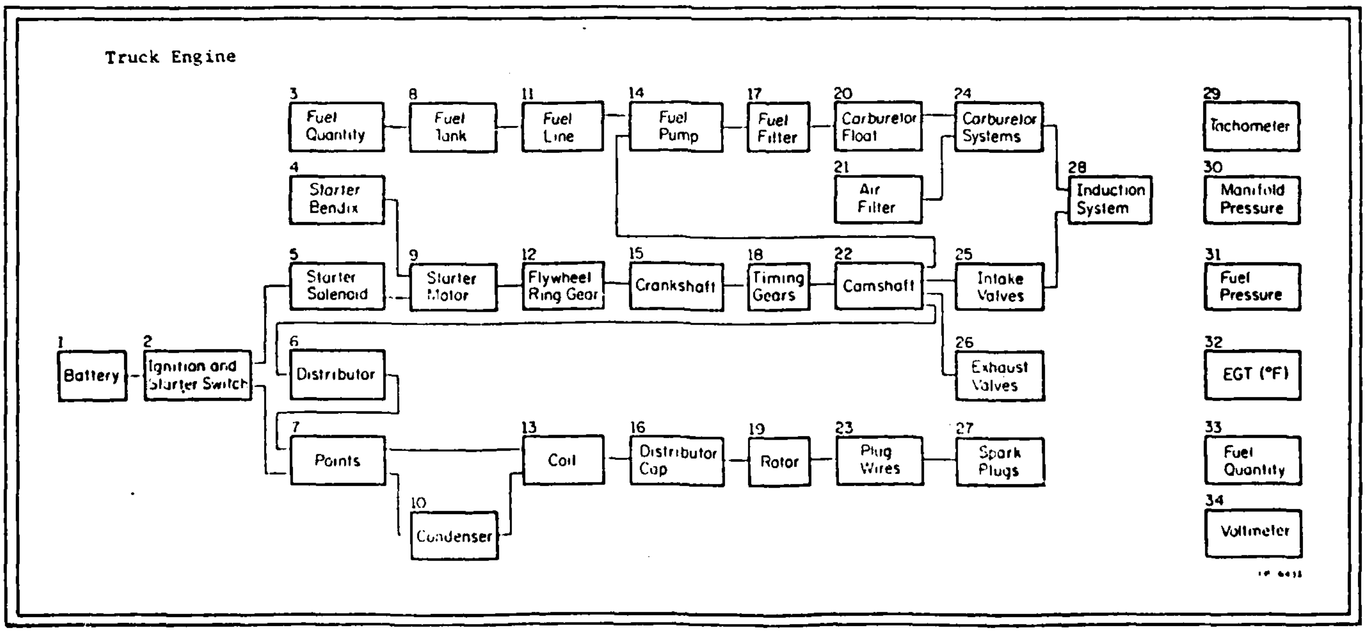

Figure 1: Hardcopy Diagram of Truck Engine for PLATO FAULT

While execution of the PLATO lesson "FAULT" is completely self-explanatory, when using a PLATO IV terminal, this section will describe each action available to the troubleshooter. The simulation's screen display is shown in Figure 2.

System: Truck Engine Problem: 13 of 28
Symptom: The engine runs rough and is hard to start.

Gauges

29	TACHOMETER	MARGINAL
30	MANIFOLD PRESSURE	MARGINAL
31	FUEL PRESSURE	MARGINAL
32	EGT	MARGINAL
33	FUEL QUANTITY	NORMAL
34	VOLTMETER	NORMAL

Options

Free		Expense	
INFORMATION	IX	OBSERVATION	OX,Y
COMPARISON	CX,Y,Z	BENCH TEST	BX
DESCRIPTION	DX	REPLACE PART	RX
GAUGES	GX	GIVE UP	?
POSSIBLE	P		

(X, Y, AND Z DENOTE PART NUMBERS.)

Actions			Costs
1	023	SPARK PLUG WIRES,	$ 2
	27	SPARK PLUGS - ABNORMAL	
2	B19	ROTOR - NORMAL	6
3	022	CAMSHAFT,	4
	6	DISTRIBUTOR - ABNORMAL	
4	018	TIMING GEARS,	35
	22	CAMSHAFT - NORMAL	
5	022	CAMSHAFT,	4
	14	FUEL PUMP - ABNORMAL	
6	REPLACE 22 CAMSHAFT		1300

Choose an option or press HELP for help r22

Right! You have replaced the failed part!

Press NEXT to see how well you did compared to others.

Figure 2: Terminal Display of PLATO FAULT

The previous experiments with FAULT clearly indicated that the simulation was most effective when it provided maximum information about the system and its associated testing procedures. Further, it seemed to be valuable to provide maximum information about the sequence of tests that had been performed. This information had to be presented in a concise screen display that would be very readable, even to the novice simulation user. For this reason, we chose to capitalize on PLATO's graphic capability and redesign the display.

The display contains several data elements. "System: Truck Engine" is listed at the top left of the display. This feature will be particularly important when other systems are in operation. Below the system name is the symptom information: "Symptom: The engine runs rough and is hard to start." At the top right of the display is the information that this is the thirteenth of twenty-eight randomly presented problems: "Problem: 13 of 28." The experimenter may choose which problems will be given. The object of the game is to find the single failed part as quickly as possible while incurring the minimum expense. We feel that the time vs. money tradeoff is representative of the real world of equipment troubleshooting.

The top left quadrant displays gauge information. In this example, all the gauges have been read. Since the troubleshooter typed "g" followed by a gauge number and repeated this action for each gauge number, the gauge readings required six separate actions which were recorded in the student data file. The lower

left quadrant of the large square in Figure 2 is the list of nine possible options. These options allow the troubleshooter to gather information about the problem, act on the information, and to receive information about the action. The list of options is divided into two sections, one with free options, the other requiring an expenditure of funds.

The first free option is "Information" (see Figure 3). This block provides information about the function of the part, the *a priori* probability of failure, and the costs associated with observation, bench testing, and replacement.

Information about Camshaft

The camshaft opens the valves in the proper order

Probability of Failure	Cost to Observe	Cost to Bench Test	Cost to Replace
.02	$4	$875	$1300

Figure 3: Information

The next free option is "Comparison." The comparison option will display any three parts with information about costs and probability of failure (see Figure 4).

Comparison

Part Number	Probability of Failure	Cost to Observe	Cost to Bench Test	Cost to Replace
1	.09	$ 4	$ 15	$ 135
5	.01	$ 2	$ 27	$ 43
15	.01	$10	$1000	$2100

Figure 4: Comparison

The "Description" option shows the methods of performing a bench test and provides specific steps for the test, including a listing of the appropriate tools (see Figure 5).

How to Bench Test Timing Gears:

Remove the radiator, water pump, harmonic balancer, and timing gear cover. Visually inspect the timing gears for mechanical wear. The chain should not be excessively loose. Check the manual for tolerances on chain and for gear backlash.

Figure 5: Bench Test

"Gauge" option has been briefly described previously and can be seen in the full panel display in Figure 2.

The last of the free options is P for "Possible." This choice provides the troubleshooter with the number of parts that remain within the feasible set of potential failed parts. The semi-intelligent feature of the simulation is based solely on the topography of the hardcopy display in light of the tests performed (see Figure 6).

Possible

Based on your previous actions, there are _____ remaining parts that could be the failed part.

Figure 6: Possible

The options listed in the "Expense" column of Figure 2 are the types of action-oriented procedures that would be accomplished by a real-world troubleshooter. The first of these actions is "Observation." This action determines the quality of the output from one part to the next. Steps 1, 3, 4 and 5 in the ACTION list of Figure 2 are examples of observations. When the observation is performed, the details of undertaking the action are depicted at the bottom of the display as shown in Figure 7.

Observation on Spark Plug Wires

The output from the spark plug wires is abnormal.

Remove the wire at the plug end. Hold the wire approximately 1/8 inch from ground and crank the engine. A crisp blue spark should jump the gap and no shock should be felt through the wires.

Figure 7: Observation

The "Bench Test" is the second expense option. When the bench test is performed, the troubleshooter essentially removes the suspect component from the system for a thorough check. Step 2 ("B19 Rotor-Normal") in the Action list of Figure 2 is an example of a bench test. Although this test is expensive due to the labor of removal, downtime, etc., it is the only sure way of establishing the functional integrity of a given component. "Replace Part" is self-explanatory. When the troubleshooter has incorrectly replaced a functioning part, he receives a statement to that effect. When replacement is appropriate, the troubleshooter is reinforced in his decision by a positive statement and is provided with a problem summary. When a replacement is preceded by a bench test of the same part the cost of replacement is modified.

The performance summary, depicted in Figure 8, includes the troubleshooter's time and cost compared to the mean values

associated with that problem. Further, the summary lists the number of unnecessary steps compared to the mean number of unnecessary, or redundant, actions performed on that problem.

	Total Test Cost	Total Time	Total Actions	Unnecessary Actions
You	$1339	1:35	7	1
Average of 5 users	$1385	1:07	10	3

Figure 8: Performance Summary

"Give up" permits the frustrated problem solver to quit a problem by typing "?". The penalty for giving up is the replacement of all system components at a cost equal to the total replacement cost of all components. PLATO FAULT adds these individual part costs in a slow running total while allowing the troubleshooter to abort this procedure at any time.

III. FAULT Documentation

The documentation for PLATO FAULT is divided into two separate sections. The first section is a manual for the user of the simulation. The second section is written for the experimenter.

A. Troubleshooter's Manual

The hardcopy information needed by the PLATO FAULT troubleshooter is limited to the system schematic. All other explanations about FAULT are presented on-line, via the HELP key. This information is included in Appendix A.

B. Researcher's Manual

A test-management index for the researcher is provided in the TUTOR program, "faultman." The options available from this index are described in an on-line HELP section, as shown in Appendix B.

IV. Project Summary and Future Research

The project has developed a PLATO-based simulation quite similar to the FAULT simulation developed by Rouse and his associates at the University of Illinois. The simulation was refined to serve the troubleshooter and researcher.

The PLATO version of FAULT makes it very easy for the researcher to set up a demonstration or an experiment. The researcher has the ability to select the problems that will be presented to the troubleshooter. Additionally, it is very easy for the researcher to access the troubleshooter data. This data can be accessed at a global level (i.e., average times and costs to problem solution) or on a very specific level (i.e., listing of each action required to solve a problem). These options are available through the "Faultman" lesson (see Appendix B).

PLATO FAULT has made a number of improvements for the troubleshooter. The display has been redesigned to provide more information in natural language. The list of options has been split to clearly differentiate between cost related and free options. Most importantly, PLATO FAULT has been enhanced with numerous on-line aids to help the troubleshooter through the dynamics of the simulation.

The suggestions for future related research were presented quite extensively in the Project Status report delivered to ARI on May 24, 1982. While the ideas presented

in that report ran an extensive gamut, it should be emphasized that the development of a PLATO FAULT editor should be of the highest ARI priority. This editor would permit ARI to develop new systems and modify existing simulations.

V. Acknowledgements

The authors would like to acknowledge the persons who contributed to this project. Dr. Martha Weller handled many of the administrative details of the contract and written reports, particularly in the final months. Dr. Henry Taylor provided invaluable assistance throughout the project. He participated in the writing of the reports and provided the positive administrative support which permitted all of us to spend a considerably large portion of time on the satisfactory completion of this contract. Further, we appreciate the evaluation and feedback provided by Dr. Douglas Bobko and Dr. Michael Drillings at the Army Research Institute.

VI. Bibliography

Hunt, R.M. A Study of transfer of problem solving skills from context-free to context-specific fault diagnosis tasks. Urbana, IL: University of Illinois, Coordinated Science Laboratory, Report No. T-82, July 1979.

Hunt, R.M. and Rouse, W.B. Problem solving skills of maintenance trainees in diagnosing faults in simulated powerplants. Human Factors, 1981, 23, (3), 317-328.

Johnson, W.B. Computer simulations for fault diagnosis training: an empirical study of learning transfer from simulation to live system performance (Doctoral Dissertation, University of Illinois). Dissertation Abstracts International, 1981, 41, (11), 4625-A. (University Microfilms No. 8108555).

Johnson, W.B. Development of a Laboratory Tool for research in the design of games for training of troubleshooting skills: project status and suggestions for future research, Interim report for contract MDA 903-82-M-3924 submitted to Dr. Douglas Bobko on May 24, 1982 (a).

Johnson, W.B. and Rouse, W. B. Analysis and classification of human errors in troubleshooting live aircraft powerplants. IEEE Transactions on Systems, Man, and Cybernetics, 1982, SMC-12, (3), May/June, 389-393 (b).

Johnson, W.B. and Rouse, W. B. Training maintenance technicians for troubleshooting: two experiments with computer simulations. Human Factors, 1982, 24, (3), 271-276 (c).

Rouse, W. B. and Hunt, R. M. Human problem solving in fault diagnosis tasks: Final report for contract MDA 903-79-C-0421, July 22, 1979 - July 21, 1982. Submitted to ARI July 21, 1982.

APPENDIX A

FAULT Help Index

1. Introduction to FAULT
2. Free Options
3. Expense Options
4. Scoring

Choose an option or press BACK to return »

Press SHIFT-BACK from anywhere in this help section to return to the help index.

<u>Introduction to FAULT</u>

At the start of each problem, you will be given a symptom of a malfunctioning system in which a single part has failed. Nine troubleshooting options are available to assist you in locating and replacing the defective part. Some of these options, such as bench testing a part or observing the output from a part, have associated costs that are comparable to those in the real world. In addition, a record will be kept of the time it takes you to solve a problem.

<u>The object of this game is to find and replace the failed part while spending a minimum amount of time and money</u>.

Press NEXT to continue or BACK to review.

Troubleshooting Options

The following troubleshooting options are available:

Free		Expense	
Information	Ix	Observation	Ox,y
Comparison	Cx,y,z	Bench Test	Bx
Description	Dx	Replace Part	Rx
Gauges	Gx	Give Up	?
Possible	P		
(x, y, and z denote part numbers)			

The options listed on the left are free, while the options on the right will have associated costs.

The letters following each option will help you remember how to request that option. For example, to request information about a part, you would type "i" followed by the part number. Use the ERASE key to make any corrections or changes. Then, press NEXT to enter your request.

Following is a description of the information that each of these options provides.

Press NEXT to continue or BACK to review.

<u>Information</u> {Ix} ** FREE **

To request information about a given part, type "i" followed by the part number. You will then receive the following information:

1) The <u>a priori</u> probability of failure
2) Observation cost
3) Bench test cost
4) Replacement cost
5) A short functional description

<u>Comparison</u> {Cx,y,z} ** FREE **

To obtain information on up to three parts at a time, type "c" followed by the numbers of the parts to be compared, separated by commas. For example, "c12,34" would give you the following information:

Part Number	Probability of Failure	Cost to Observe	Cost to Bench Test	Cost to Replace
12	.02	$2	$35	$135
34	.04	$1	$25	$40

Press NEXT to continue or BACK to review.

Description {Dx} ** FREE **

For a description of how to perform a bench test on a part, type "d" followed by the part number. This option also tells you what tools and manuals are necessary to perform the test in the real world.

Gauge Reading {Gx} ** FREE **

In an effort to gather information about the status of the system, you may choose to observe the reading on a gauge. To do this, type "g" followed by the number of the gauge. You will receive one of the following responses:

1) Normal
2) High
3) Low
4) Marginal
5) Variable
6) Zero

You will not be charged for this information, but there is no reason to check gauges unecessarily.

Press NEXT to continue or BACK to review.

Possible {P} ** FREE **

Type "p" for this option, which will help you narrow down the number of parts that could possibly be the defective part. Without regard to the symptom, PLATO will look at the observations and bench tests you have made and determine how many parts in the system could possibly be the failed part.

Press NEXT to continue or BACK to review.

Observation {Ox,y} $$ EXPENSE $$

An observation is made between two parts to see if the output from the first part to the second part is acceptable. To make an observation, type "o", then the number of the first part, then a comma, then the number of the second part. For example, "o1,2" would give you one of the following responses:

1)	Normal	The output of the first part is acceptable.
2)	Abnormal	The output of the first part is unacceptable. Either the first part has failed or one of its inputs is unacceptable.
3)	Marginal	The output of the first part is marginal. Either the first part is weak or one of its inputs is weak.
4)	Null	There is no connection between the parts you have selected.
5)	Unavailable	This observation cannot be made due to the specific nature of the parts involved.

Press NEXT to continue or BACK to review.

Observation {Ox,y} (continued) $$ EXPENSE $$

Remember that the purpose of making an observation is to see if the output from the first part to the second part is acceptable; therefore, when typing in the part numbers, ORDER IS IMPORTANT. You can also observe a terminal part (a part that supplies no output to another part) to see if it is working properly. To do this, type "o" followed by the part number.

You will be charged for each observation that you make according to the time and effort needed to make such an observation.

Press NEXT to continue or BACK to review.

Bench Test {Bx} $$ EXPENSE $$

This test involves removing a part from the system and testing that part independently of the rest of the system. To do this, type "b" followed by the number of the part to be tested. You will then receive one of the following responses:

1) Normal — The part is working normally.

2) Abnormal — The part has failed completely and should be replaced.

3) Marginal — The part has failed marginally and should be replaced.

You will be charged for bench tests, and they do tend to be expensive, but their results are conclusive and should help to avoid unnecessary replacements.

Press NEXT to continue or BACK to review.

Replace {Rx} $$ EXPENSE $$

When you think you have located the failed part, you should replace that part. To do this, type "r" followed by the part number. You will be charged for the cost of the part and the time to replace it. If the replacement immediately follows a bench test of the same part, you will be charged the difference between the two action costs. In other words, the bench test cost will be counted towards the cost of replacement.

Give Up {?} $$ EXPENSE $$

As a last resort, if you feel that you cannot solve the problem, this option will allow you to go on to the next problem. To do this, type "?". You should avoid this action, however, because you will be charged for replacing every part in the entire system.

Press NEXT to continue or BACK to review.

FAULT Scoring

After you have correctly solved a problem, you will be given a recap of what you did and how much you spent before going on to the next problem. To give you an idea of how your performance compares to that of others who have solved the problem, the following information will be displayed:

	Total Cost	Total Time	Total Actions	Unnecessary Actions
You	$19	1:35	7	2
Average of 271 users	$27	1:07	10	5

To avoid inflation of these averages, the results of the "Give Up" option are not included.

If you cannot solve a problem in 60 actions or less, you will automatically go on to the next problem.

Press NEXT to continue or BACK to review.

APPENDIX B

FAULT Test Manager Help

The FAULT Test Manager lets you collect detailed performance data on users in any one PLATO group. The problems which are to be presented can be selected ahead of time instead of being randomly chosen from all possible problems, as is done in the regular, non-testing mode of operation. There is a limit of 100 users for which data can be stored at one time. Before you give a test, you must make sure that there is sufficient space available to store the results of all those who are to take the test. To make room for new results, you must delete some or all of the old results via options in this lesson (see below). You might want to make a hardcopy of the results or transfer them to another file before deleting them. So, a typical set of steps in the administration of a test would include:

1) Copying the previous test results.
2) Deleting the previous test results.
3) Specifying the test group.
4) Running the users through FAULT.
5) Clearing the test group.

A description of each of the options available to you from the FAULT Test Manager index follows:

Press NEXT to continue or BACK for the index.

FAULT Test Manager

1. Specify test group (currently "arl")

2. Select test problems (5 selected)

3. Inspect/edit test results

4. Inspect/edit problem statistics

5. Delete all test results

Choose an option or press HELP for help ≫

1. Specify test group

This option allows you to specify the PLATO group for which individual test data are to be collected. To turn off the collection of test data, clear the specified test group by pressing SHIFT-HELP. The test results remain available until specifically deleted via this lesson.

2. Select test problems

You can choose which problems are to be presented to the test group. Note that each problem corresponds to one failed part in the system and its associated symptom and gauge readings. To see the failure information for a given part, enter the part number and press NEXT. You can then add the problem to the test by pressing LAB, or if the problem is already selected, you can delete it from the test by pressing SHIFT-HELP. Each problem can be selected only once and there is a maximum of 30 problems that can be selected for any one test. During the test, the selected problems are presented in a random order.

Press NEXT to continue or BACK to review.

FAULT Test Manager

1. Specify test group (currently "arl")

2. Select test problems (5 selected)

3. Inspect/edit test results

4. Inspect/edit problem statistics

5. Delete all test results

Enter the group to be tested ≫

Press SHIFT-HELP to clear it.

FAULT Test Editor
(5 problems selected)

2 ignition/starter switch
5 starter solenoid
7 points
10 condensor
23 spark plug wires

Part 11: fuel line

Symptom: The engine runs rough.

Gauge Readings:

29	30	31	32	33	34				
var	high	low	high	norm	norm				

Press LAB to add this problem or BACK to select another.

3. Inspect/edit test results

To inspect an individual's test results, enter the user's name and group, then press NEXT. You have the option to delete the user's test results at this time by pressing SHIFT-HELP. If you prefer, you can get a listing of all the users tested by pressing DATA, from which you can choose the one you want. The test data for each problem for each user consists of: the failed part; the starting and finishing times; the total time; the total cost; and the total number of actions and unnecessary actions. Also displayed for each action within each problem are the action taken, the parts involved, the result of the action, the cost, the time, and whether or not the action was necessary.

Press NEXT to continue or BACK to review.

FAULT Test Results

Data stored for 6 students.

Enter name/group   entwistle/arl

Press NEXT to see records,
SHIFT-HELP to delete records, or
BACK to select another name.

FAULT Test Results

1 entwistle/arl
2 johnson/arl
3 ksg/arl
4 taylor/arl
5 van proyen/arl
6 ziegler/arl

Enter a number 1

Press NEXT to see records,
SHIFT-HELP to delete records, or
BACK to choose another number.

Name: entwistle/arl

Problem: 2 of 5

Failure: condensor
Started: 08/27/82 at 09:12:04
Finished: 08/27/82 at 09:14:37

Total time: 1:50
Total cost: $13
Unnecessary actions: 0

Action	Time	Cost	Action	Time	Cost
1 g33 norm	0:02	$			$
2 g31 norm	0:03				
3 c5,9,12	0:20				
4 o9,12 norm	0:05	1			
5 o7,13 norm	0:09	1			
6 o27,27 abno	0:28	2			
7 i13	0:12				
8 o13,16 abno	0:09	2			
9 o10,13 abno	0:10	2			
10 replace 10	0:13	5			

Press + for the next problem, - for the previous one,
or press a number key to advance that many problems.

Press SHIFT-NEXT or SHIFT-BACK to move between students.

4. Inspect problem statistics

The problem statistics include the average time, average cost, and average number of actions and unnecessary actions needed to complete each problem. Note that these averages are computed from all users of FAULT, rather than from the current test group or from the test results currently stored. For this reason, the number of users to complete each problem is included in the statistics display. You can initialize all the problem statistics by pressing SHIFT-HELP on the statistics display page.

5. Delete all test results

This option permanently deletes all of the test results currently stored. The problem statistics remain unchanged. To delete the test results of an individual user, use the "inspect/edit test results" option. Test results should be deleted whenever a new group is to be tested unless you want to combine the results of more than one group. Remember that there is a limit on the number of users whose test results can be stored at one time.

Press NEXT for the index or BACK to review.

Average Problem Statistics

	Part Name	Users	Total Cost	Total Time	Total Actions	Unneeded Actions
1	battery	11	$ 155	0:43	6	0
2	ignition/starter switch	13	80	0:44	8	0
3	fuel quantity	10	93	0:23	5	0
4	starter bendix	10	655	1:33	11	0
5	starter solenoid	18	154	1:16	10	0
6	distributor	8	406	1:06	16	0
7	points	6	40	0:49	10	0
8	fuel tank	4	492	0:38	7	0
9	starter motor	3	286	0:43	12	0
10	condensor	12	44	0:56	11	0
11	fuel line	10	774	1:18	15	0
12	flywheel ring gear	10	1454	3:57	13	0
13	coil	6	135	0:52	15	0
14	fuel pump	8	191	0:45	12	0
15	crankshaft	11	2783	1:11	18	0
16	distributor cap	7	451	2:05	16	0
17	fuel filter	8	102	1:06	10	0
18	timing gears	6	1277	1:32	15	0
19	rotor	5	45	0:44	11	0
20	carburetor float	4	770	0:57	16	0

Press NEXT for more statistics.

Press SHIFT-HELP to zero ALL the problem statistics.

Are you sure??

If so, press SHIFT-HELP to delete all
the test results currently stored.